Reprinted from JOURNAL OF COMPUTATIONAL PHYSICS
All Rights Reserved by Academic Press, New York and London

Vol. 90, No. 2, October 1990
Printed in Belgium

A Second-Order Implicit Particle Mover with Adjustable Damping

ALEX FRIEDMAN

Lawrence Livermore National Laboratory, University of California, Livermore, California 94550

Received April 8, 1989; revised September 19, 1989

A new algorithm for the calculation of particle trajectories is introduced. The algorithm combines second-order accuracy in the real frequency with third-order user-adjustable attenuation. It requires little storage of data from previous time levels. The method was designed for use in implicit particle-in-cell plasma simulation codes, and this application is treated in detail. It may also prove useful in other applications where one seeks to preserve the accuracy of low-frequency oscillations while rapidly damping under-resolved high frequency motions, e.g., solution of the field equations in electromagnetic particle codes. An explicit variation, wherein future quantities are obtained by extrapolation, can provide attenuation but not large-timestep stability. © 1990 Academic Press, Inc.

I. INTRODUCTION

An implicit particle-in-cell simulation should ideally employ a time advancement scheme which accurately follows motions which are well resolved while rapidly damping under-resolved modes. Storage limitations and algorithmic complexity generally preclude use of a high-order difference scheme. Indeed, schemes in common use include one which is almost time-centered but employs a small bias toward backward time differencing [1]; this leads to a relative damping rate γ/ω_0 which is proportional to $\omega_0 \Delta t$, where Δt is the timestep size and ω_0 the analytical mode frequency. Thus, the scheme tends to be somewhat dissipative even for very low frequencies. Another popular algorithm, the *d1 scheme* [2], imposes damping which scales as the cube of the timestep size; however, it has no adjustable parameters, and so the user has no control over the amount of damping introduced except by varying the timestep size. The damping may be appreciable for modes with $\omega_0 \Delta t \lesssim 1$, even though these modes are "resolved" with ten or more steps per orbit. In some cases this fixed level of damping may be quite large [3]. If the simulation parameters are not carefully chosen or the system is sufficiently inhomogeneous, an overly rapid decay of phase space structure may result, so that collective behaviors of real interest are not observable.

Here a new time advancement algorithm is introduced. The algorithm combines third-order damping with a user-adjustable attenuation. Formally, it is a "blend" of

0021-9991/90 $3.00

the $d1$ and $c0$ schemes [2], with the algorithmic steps arranged so as to require no storage above that already required by the $d1$ scheme. Codes which employ the $d1$ scheme, such as TESS [4] and AVANTI [5], can be readily modified to incorporate this generalized algorithm. In Section II below, the basic algorithm is introduced. Two variants are presented, one with position x and velocity v offset from each other in time by a half step, the other with them both centered at the same time levels. In Section III, a practical implementation with offset centering is presented, and in Section IV a similar scheme with integral centering is outlined.

Also derived (in Sections V and VI) are approximate and exact dispersion relations for motion in a harmonic well. It is shown that the $d1$ scheme, as it is usually implemented, admits numerical instability in such a well for large enough timestep. In self-consistent direct-implicit plasma simulation codes [6–8] this instability is generally suppressed by the reduction in field magnitude arising from the effective susceptibility used; however, the suppression may be incomplete under certain circumstances.

Tests of the new mover, as applied to motion in a fixed potential, appear in Section VII, and full direct-implicit simulations are exhibited in Section VIII. In Section IX, procedures for non-constant timestep and damping parameter are presented. Finally, Section X contains a discussion of some implications of this work. That last section also outlines how the method may be applied to the solution of Maxwell's equations and briefly describes an explicit variant.

II. The Basic Algorithm

The most elementary form of the algorithm applies to a second-order differential equation in the position x of a particle; it makes no reference to a velocity. The equation to be solved is

$$\frac{d^2x(t)}{dt^2} = a(x(t), t). \tag{1}$$

One begins with a conceptual implicit difference equation which employs the acceleration $a(x, t)$ at the advanced time level without specifying how it is obtained. Denoting time levels by subscripts, the algorithm is

$$\frac{x_{n+1} - 2x_n + x_{n-1}}{\Delta t^2} = \frac{a_{n+1} + \bar{A}_{n-1}}{2}, \tag{2}$$

where

$$\bar{A}_{n-1} \equiv (\theta/2)\, a_n + (1 - \theta/2)\, \bar{a}_{n-2} \tag{3a}$$

$$\bar{a}_{n-1} \equiv (1 - \theta/2)\, a_n + (\theta/2)\, \bar{a}_{n-2}. \tag{3b}$$

Here a_n is the acceleration obtained by evaluating the force field at x_n, the particle's position at time level n. In an unmagnetized one-dimensional particle-in-cell plasma simulation, $a_n \equiv (q/m)\, E_n(x_n)$, with q the particle charge, m the mass, and E the electric field. The $\bar{a}$'s are lag-averaged accelerations. The $\bar{A}$'s are temporary quantities defined for convenience. θ is the user-specified damping parameter; for $\theta = 1$ the $d1$ scheme of [2] is recovered, while for $\theta = 0$ the reversible, undamped $c0$ scheme (with $c_0 = \frac{1}{2}$) results.

From this form one can derive several possible sets of first-order equations which incorporate a velocity. Perhaps the most familiar form involves offset x and v in an expression reminiscent of the "leapfrog" advance:

$$v_{n+1/2} = v_{n-1/2} + (\Delta t/2)[a_{n+1} + \bar{A}_{n-1}] \tag{4a}$$

$$x_{n+1} = x_n + \Delta t\, v_{n+1/2}. \tag{4b}$$

In the absence of a magnetic force, the velocity is just a "bookkeeping" quantity which allows one to split the single second-order equation (2) into a pair of first-order equations. Equations (4) are algebraically identical to Eq. (2); to see this, shift Eq. (4b) by one time level and subtract from the original, then use Eq. (4a) to eliminate the velocities from the result.

One can readily show that the above scheme is formally "time-centered;" that is, that the acceleration term in the right member of (2) or (4a) is defined at time level (abbreviated hereafter as "tl") n. Since it is an average of a_{n+1} (which is clearly a tl $n+1$ quantity) and $\bar{A}_{n-1}$, one need only show that the latter is truly a tl $n-1$ quantity. To see this, begin with Eq. (3b); one writes

$$\mathrm{tl}(\bar{a}_{n-1}) = (1-\theta/2)[n] + (\theta/2)[\mathrm{tl}(\bar{a}_{n-1}) - 1]. \tag{5}$$

Solving, one obtains

$$\mathrm{tl}(\bar{a}_{n-1}) = n - (\theta/2)/(1-\theta/2). \tag{6}$$

Then,

$$\begin{aligned} \mathrm{tl}(\bar{A}_{n-1}) &= (\theta/2)[n] + (1-\theta/2)[n - (\theta/2)/(1-\theta/2) - 1] \\ &= n - 1, \end{aligned} \tag{7}$$

which is the desired result.

One can also write the adjustable mover in a form wherein both x and v are centered at integral times. The algorithm becomes

$$v_{n+1} = v_n + (\Delta t/2)[a_{n+1} + \bar{A}_n] \tag{8a}$$

$$x_{n+1} = x_n + \Delta t\, v_n + (\Delta t^2/2)\, a_{n+1}. \tag{8b}$$

This scheme is also algebraically equivalent to Eq. (2); it too is formally time centered, since $v_n + (\Delta t/2)\, a_{n+1}$ is a tl $n+1/2$ quantity (for constant acceleration).

The above schemes generalize to higher dimensionality in a straightforward manner. To introduce a magnetic field in the offset mover one can use a vector form of Eq. (4b) along with a generalization of Eq. (4a),

$$\mathbf{v}_{n+1/2} = \mathbf{v}_{n-1/2} + (\Delta t/2)[\mathbf{a}_{n+1} + \bar{\mathbf{A}}_{n-1} + (\mathbf{v}_{n+1/2} + \mathbf{v}_{n-1/2}) \times \Omega_n], \tag{9}$$

where $\Omega_n \equiv q\mathbf{B}_n(\mathbf{x}_n)/mc$ is the (vector) gyration frequency. The magnetic rotation in Eq. (9) is undamped but implicit; except for phase errors it yields the correct gyration for arbitrarily large timestep. In most circumstances this behavior is preferable to decaying gyro-motion.

One can transform this into a scheme with an integral velocity, which is defined by the half-advance:

$$\mathbf{v}_n = \mathbf{v}_{n-1/2} + (\Delta t/2)[\bar{\mathbf{A}}_{n-1} + \mathbf{v}_{n-1/2} \times \Omega_n]. \tag{10}$$

Subtracting Eq. (10) from Eq. (9) one obtains

$$\mathbf{v}_{n+1/2} = \mathbf{v}_n + (\Delta t/2)[\mathbf{a}_{n+1} + \mathbf{v}_{n+1/2} \times \Omega_n], \tag{11}$$

and re-writing Eq. (10) shifted by one time level, one obtains a pair of steps which replace Eq. (8a), namely Eq. (11) followed by

$$\mathbf{v}_{n+1} = \mathbf{v}_{n+1/2} + (\Delta t/2)[\bar{\mathbf{A}}_n + \mathbf{v}_{n+1/2} \times \Omega_{n+1}]. \tag{12}$$

III. A Practical Difference Scheme: Offset Centering of x and v

A first practical simulation algorithm employs staggered time levels for x and v. To derive a practical mover suitable for (e.g.) direct-implicit plasma simulation one seeks to separate the effects of the future electric field from those of known fields and to employ a velocity advance to a $\tilde{\mathbf{v}}$ which involves only explicitly known quantities. Setting $\mathbf{a}_{n+1}$ to zero in Eq. (9) leads to

$$\tilde{\mathbf{v}}_{n+1/2} = \mathbf{v}_{n-1/2} + (\Delta t/2)[\bar{\mathbf{A}}_{n-1} + (\tilde{\mathbf{v}}_{n+1/2} + \mathbf{v}_{n-1/2}) \times \Omega_n], \tag{13}$$

and subtracting Eq. (13) from Eq. (9) gives

$$\delta\mathbf{v}_{n+1/2} \equiv \mathbf{v}_{n+1/2} - \tilde{\mathbf{v}}_{n+1/2} = (\Delta t/2)[\mathbf{a}_{n+1} + \delta\mathbf{v}_{n+1/2} \times \Omega_n], \tag{14}$$

This apportionment of the electric acceleration between the two velocity-advance steps assures that $x_{n+1} = x_n + \Delta t\, v_{n+1/2}$, i.e., that $v_{n+1/2}$ is a true half-integral velocity. This makes incorporation of a magnetic field straightforward: only the lagged electric force differs from (e.g.) those of Ref. [5], and the method presented there is directly applicable.

With this information one can write the complete algorithm. One enters a step with particle quantities $\tilde{\mathbf{v}}_{n-1/2}$, $\tilde{\mathbf{x}}_n$, $\tilde{\mathbf{a}}_{n-2}$, Ω_{n-1}, and with $\mathbf{E}_n$, $\mathbf{B}_n$ on the mesh.

ALGORITHM 1.

The final-push is:

(1) $\mathbf{a}_n = (q/m)\,\mathbf{E}_n(\tilde{\mathbf{x}}_n)$ (interpolation of field from mesh)

(2) $\delta\mathbf{v}_{n-1/2} = (\Delta t/2)[\mathbf{a}_n + \delta\mathbf{v}_{n-1/2} \times \Omega_{n-1}]$

(3) $\mathbf{v}_{n-1/2} = \tilde{\mathbf{v}}_{n-1/2} + \delta\mathbf{v}_{n-1/2}$

(4) $\mathbf{x}_n = \tilde{\mathbf{x}}_n + \Delta t\,\delta\mathbf{v}_{n-1/2}$

(5) $\bar{\mathbf{A}}_{n-1} = (\theta/2)\,\mathbf{a}_n + (1-\theta/2)\,\bar{\mathbf{a}}_{n-2}$

(6) $\bar{\mathbf{a}}_{n-1} = (1-\theta/2)\,\mathbf{a}_n + (\theta/2)\,\bar{\mathbf{a}}_{n-2}$

(7) $\Omega_n = (q/mc)\,\mathbf{B}_n(\mathbf{x}_n)$ (interpolation of field from mesh).

The pre-push is:

(8) $\tilde{\mathbf{v}}_{n+1/2} = \mathbf{v}_{n-1/2} + (\Delta t/2)[\bar{\mathbf{A}}_{n-1} + (\tilde{\mathbf{v}}_{n+1/2} + \mathbf{v}_{n-1/2}) \times \Omega_n]$

(9) $\tilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \Delta t\,\tilde{\mathbf{v}}_{n+1/2}$.

If a direct-implicit field solution is to be carried out, the "extrapolated" or "free-streaming" quantities $\tilde{\mathbf{v}}_{n+1/2}$ and $\tilde{\mathbf{x}}_{n+1}$ provide the information needed for the source terms, as well as the effective susceptibility and magnetization. Inversion of 3×3 matrices is required in steps (2) and (8).

When $\theta = 1$ the $d1$ scheme is recovered, while when $\theta = 0$ the resulting algorithm is approximately time-centered and reversible and can thus (within that approximation) introduce no damping. The scheme would be completely reversible if (a) the electric field were evaluated at integral positions x_n rather than extrapolated positions $\tilde{x}$, and (b) the implicit field solution yielded a field that exactly corresponded to sources at particle positions x_{n+1}; thus exact reversibility would be impractical, but measure (a) might reasonably be implemented.

IV. A PRACTICAL DIFFERENCE SCHEME: INTEGRAL CENTERING OF x AND v

It is convenient to employ an implicit particle advance with all key quantities defined at integral (not staggered) time levels. This simplifies the initialization and diagnostic parts of the code and opens the possibility of varying the timestep size "between steps" in a multi-scale algorithm [9]. The Appendix contains a derivation of a $d1$ algorithm with integral centering.

One enters a timestep with particle data $\mathbf{x}_{n-1}$, $\mathbf{v}_{n-1}$, $\bar{\mathbf{a}}_{n-2}$, Ω_{n-1}, and with $\mathbf{E}_n$, $\mathbf{B}_n$ on the mesh.

ALGORITHM 2.

The "final-push" is:

(1) $\tilde{\mathbf{x}}_n = \mathbf{x}_{n-1} + \Delta t\,\mathbf{v}_{n-1}$

(2) $\mathbf{a}_n = (q/m)\,\mathbf{E}_n(\tilde{\mathbf{x}}_n)$ (interpolation of field from mesh)

(3) $\mathbf{v}_{n-1/2} = \mathbf{v}_{n-1} + (\Delta t/2)[\mathbf{a}_n + \mathbf{v}_{n-1/2} \times \Omega_{n-1}]$

(4) $\mathbf{x}_n = \mathbf{x}_{n-1} + \Delta t\, \mathbf{v}_{n-1/2}$

(5) $\bar{\mathbf{A}}_{n-1} = (\theta/2)\, \mathbf{a}_n + (1-\theta/2)\, \bar{\mathbf{a}}_{n-2}$

(6) $\bar{\mathbf{a}}_{n-1} = (1-\theta/2)\, \mathbf{a}_n + (\theta/2)\, \bar{\mathbf{a}}_{n-2}$

(7) $\Omega_n = (q/mc)\, \mathbf{B}_n(\mathbf{x}_n)$ (interpolation of field from mesh)

(8) $\mathbf{v}_n = \mathbf{v}_{n-1/2} + (\Delta t/2)[\bar{\mathbf{A}}_{n-1} + \mathbf{v}_{n-1/2} \times \Omega_n]$.

The pre-push is:

(9) $\tilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \Delta t\, \mathbf{v}_n$.

A single 3×3 matrix inversion is required, in step (3). In this algorithm we have not calculated a $\tilde{\mathbf{v}}$; such a step may be necessary if a self-consistent magnetic field is to be included.

V. Approximate Dispersion Relation

An approximate dispersion relation for this algorithm as applied to motion in a harmonic well can be derived by Fourier transforming in time. The inexactitude arises from the assumption (as in Ref. [2]) that E_n is interpolated at x_n rather than at $\tilde{x}_n$, that is, $a_n = -\omega_0^2 x_n$ rather than $a_n = -\omega_0^2(x_{n-1} + \Delta t\, v_{n-1})$. One then employs the single-step operator $z \equiv e^{-i\omega\,\Delta t}$, so that $x_n = x_0 z^n$ etc.; here ω is the mode frequency obtained numerically. From Eq. (2) we have

$$x_n - 2x_{n-1} + x_{n-2} = \frac{\Delta t^2}{2}\{a_n + (\theta/2)\, a_{n-1} + (1-\theta/2)\, \bar{a}_{n-3}\}. \tag{15}$$

Use of the recurrence relation for $\bar{a}$ enables one to rewrite this as

$$\frac{x_n - 2x_{n-1} + x_{n-2}}{\Delta t^2}$$

$$= \frac{1}{2}\{a_n + (\theta/2)\, a_{n-1} + (1-\theta/2)^2\, [a_{n-2} + (\theta/2)\, a_{n-3} + (\theta/2)^2\, a_{n-4} + \cdots]\}. \tag{16}$$

Substituting $a_n = -\omega_0^2 x_n$ etc. as mentioned above, then using $x_n = x_0 z^n$ etc., and finally summing the power series in $(\theta/2)\, z^{-1}$, the dispersion relation is

$$(z-1)^2\,(2z-\theta) = -\omega_0^2\, \Delta t^2\, z(z^2 + 1 - \theta). \tag{17}$$

The dispersion relation for the $d1$ scheme was presented in [2] and acts as a useful check on this result. When θ is zero the equation becomes quadratic, $|z| \equiv 1$, and the dispersion relation becomes

$$\omega\, \Delta t = \arctan[\omega_0\, \Delta t(1 + \omega_0^2\, \Delta t^2/4)^{1/2}]; \tag{18}$$

there is no damping whatsoever.

Following a procedure similar to that used in Ref. [2], one can determine the small-timestep behavior. The full dispersion relation is

$$\frac{1}{[(2/\omega_0\,\Delta t)\sin(\omega\,\Delta t/2)]^2}=\frac{2e^{-i\omega\,\Delta t}-\theta}{e^{-2i\omega\,\Delta t}+1-\theta}. \tag{19}$$

Expanding both left and right members, this becomes

$$\omega_0^2/\omega^2+(1/3)(\omega_0\,\Delta t/2)^2=1+(\omega\,\Delta t)^2/(2-\theta)+i(\omega\,\Delta t)^3\,\theta/(2-\theta)^2+\cdots. \tag{20}$$

Finally, one obtains

$$\omega_r/\omega_0=1-\omega_0^2\,\Delta t^2[1/(4-2\theta)-1/24]+\cdots; \tag{21a}$$

$$\gamma/\omega_0=-\theta/[2(2-\theta)^2]\,\omega_0^3\,\Delta t^3+\cdots. \tag{21b}$$

There is less of a frequency reduction for the centered scheme than for the $d1$ scheme, and (as expected) no damping.

The cubic equation (17) has been solved numerically to obtain dispersion curves. The results for the least-damped roots (these are a conjugate pair) are shown in Figs. 1 and 2; the third root is purely (and strongly) damped. In Fig. 1a $|z|$ is

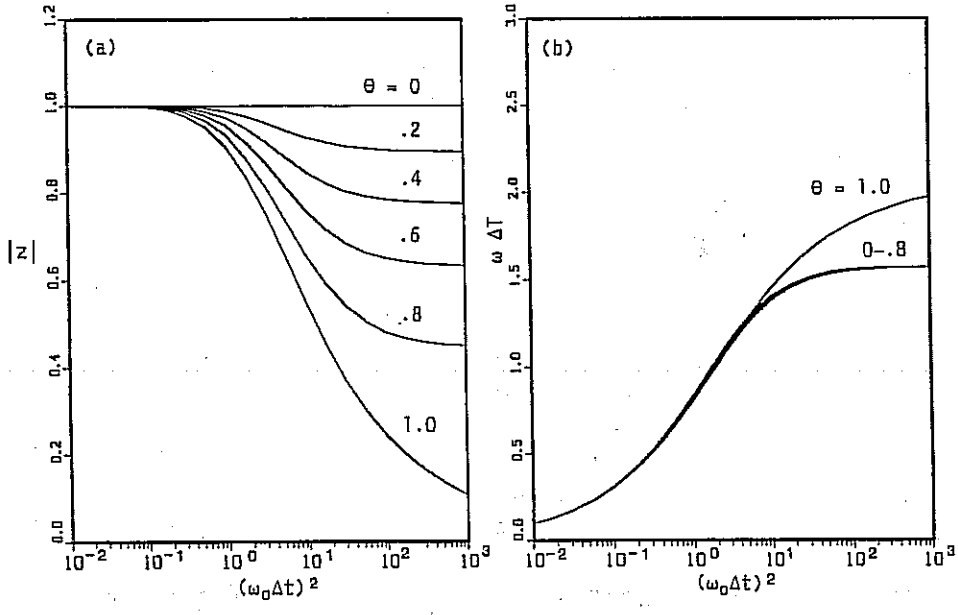

FIG. 1. Dispersion curves for the adjustable mover assuming interpolation at true future position. Plotted versus $\omega_0^2\,\Delta t^2$ are (a) $|z|$, for (from top to bottom) $\theta=0$, 0.2, 0.4, 0.6, 0.8, 1.0; (b) $\omega_r\,\Delta t$—the upper curve is $\theta=1.0$, all others fall within the lower curve.

plotted as a function of $(\omega_0^2\,\Delta t)^2$; $|z|$ is just $e^{\gamma\,\Delta t}$, and represents the fractional decay of the oscillation during a single step. The various curves represent different values of θ; from top to bottom, they are: $\theta = 0$, 0.2, 0.4, 0.6, 0.8, 1.0. In general, $|z|$ asymptotes to $\sqrt{(1-\theta)}$; for the $d1$ scheme $|z| \sim (\omega_0\,\Delta t)^{-2/3}$ as it approaches zero. Figure 1b depicts the real frequency ω_r (actually the phase angle in radians due to one step, $\omega_r\,\Delta t$) as a function of the same parameter. It was obtained using $\omega_r\,\Delta t = \arctan[-\mathrm{Im}(z)/\mathrm{Re}(z)]$. The top curve represents the $d1$ scheme, $\theta = 1.0$; the lower (rather thick) curve represents all of the other values of θ in the above list. The dependence of ω_r upon θ in this figure appears to be discontinuous; indeed, for the $d1$ scheme the asymptotic value of $\omega_r\,\Delta t$ (as $\omega_0\,\Delta t \to \infty$) is $2\pi/3$, while for values of θ less than unity the asymptote is $\pi/2$.

Figure 2 is the same type of plot, but for values (from top to bottom in 2a) of $\theta = 0.8$, 0.9, 0.96, 0.98, 0.988, 0.996, and 1.0. The same values appear in the curves in Fig. 2b but from bottom to top. These curves reveal that the transition, at large but finite $\omega_0\,\Delta t$, to the $d1$ behavior, is indeed smooth. As the value of θ is increased toward unity, the maximum of $\omega_r\,\Delta t$ occurs at larger $\omega_0\,\Delta t$.

From these figures it is clear that the damping can be made arbitrarily small, or as large as that of the $d1$ scheme, through choice of θ. However, some choices of θ yield anomalous dispersion. While $\theta = 1$ ($d1$ scheme) and $0 \leqslant \theta \lesssim 0.8$ yield a monotone real frequency as a function of the harmonic-well frequency, values in the range $0.8 \lesssim \theta \lesssim 1.0$ lead to regimes wherein the observed frequency decreases as the

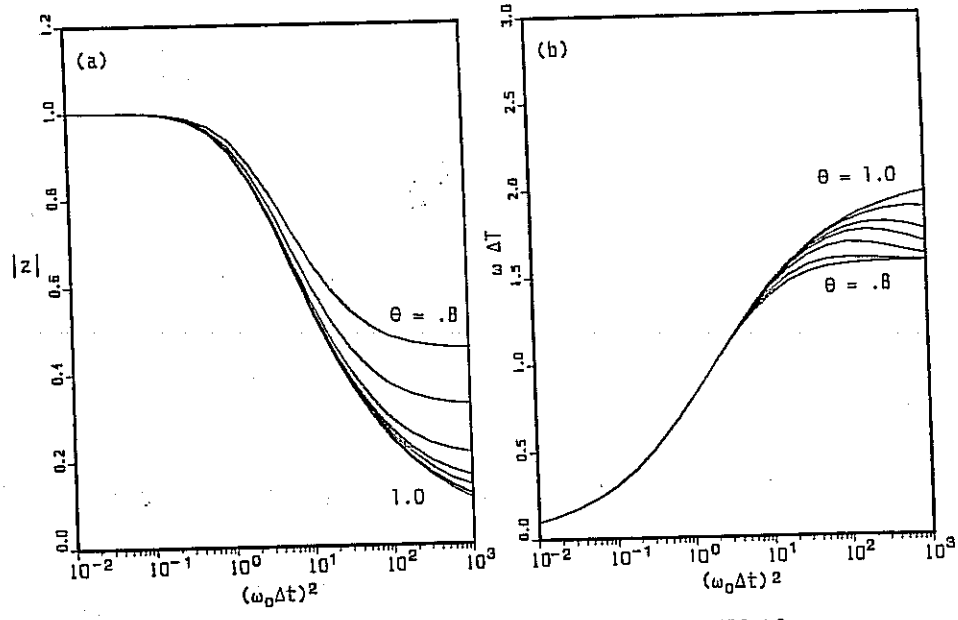

FIG. 2. As in Fig. 1, but $\theta = 0.8$, 0.9, 0.96, 0.98, 0.988, 0.996, 1.0.

harmonic-well frequency increases. These regimes are heavily damped and are unlikely to cause real difficulty in practice.

It is possible to employ values of $\theta > 1$. For $1 < \theta \leqslant 2$, z asymptotes to $\sqrt{(\theta - 1)}$, and so at large timestep the damping decreases with θ. When $\theta > 2$, $|z|$ asymptotes to a number greater than unity, and so such values are probably not useful. However, when $\theta > 1$ the least-damped root (in plots such as those above) does not exhibit smooth behavior; as $\omega_0 \Delta t$ is increased beyond a certain critical value, a different branch takes over as the least-damped root. For example, at $\theta = 3/2$ the transition occurs at $\omega_0 \Delta t \gtrsim 1$, and beyond that point $|z| \approx 0.707$ and $\omega_r = 0$. The details of the dispersion should matter little in regimes where the damping is strong.

This dispersion relation indicates global stability for all Δt. In the following section it is shown that such stability is a consequence of assuming that the future force is interpolated at the true future position. In a typical implemention the $d1$ mover and the adjustable mover admit numerical instability in a fixed well.

VI. Exact Dispersion Relation

It is also possible to obtain an exact dispersion relation for motion in a harmonic well using the mover presented here, i.e., with interpolation of the field at $\tilde{x}$. The algorithm employs

$$a_n = -\omega_0^2 \tilde{x}_n = -\omega_0^2 (x_n - \Delta t^2 a_n/2), \tag{22}$$

and solving for a_n, one obtains

$$a_n = -[\omega_0^2/(1 - \omega_0^2 \Delta t^2/2)] x_n. \tag{23}$$

Thus one need only make the replacement

$$\omega_0^2 \Delta t^2 \to W_0^2 \Delta t^2 \equiv \frac{\omega_0^2 \Delta t^2}{1 - \omega_0^2 \Delta t^2/2} \tag{24}$$

everywhere the quantity on the left appears, to obtain the exact relation. For example, the centered scheme ($\theta = 0$) has the exact dispersion relation:

$$\omega \Delta t = \arctan[\omega_0 \Delta t (1 - \omega_0^2 \Delta t^2/4)^{1/2}/(1 - \omega_0^2 \Delta t^2/2)]. \tag{25}$$

For $\omega_0 \Delta t = 1$ this gives $\omega \Delta t = \pi/3$, for a period of six timesteps.

W_0 is monotone in ω_0 until $\omega_0 \Delta t = \sqrt{2}$, where $W_0 \Delta t$ becomes infinite; the schemes begin to exhibit undesirable behavior beyond this point. The centered scheme remains neutrally stable, with $\omega_r \Delta t$ increasing to π, until $\omega_0^2 \Delta t^2 = 4$. When $\omega_0^2 \Delta t^2 = 4$, the difference scheme is: $x_{n+1} - 2x_n + x_{n-1} = 2(x_{n+1} + x_{n-1})$; the motion admits a linear-in-time divergence of the orbit point, in a sequence such as $x_0 = 1$, $x_1 = -2$, $x_2 = 3$, ... Beyond $\omega_0^2 \Delta t^2 = 4$ the centered scheme exhibits

odd–even instability. The $d1$ scheme goes odd–even unstable somewhat earlier: for $\omega_0^2 \Delta t^2 < 2$ there are damped oscillations; at 2, z is zero; between 2 and 2.4, the mode is odd–even but damped; at 2.4, there is odd–even neutral stability; and above 2.4, there is instability.

In contrast with its instability in a harmonic well, the $d1$ mover (with interpolation at $\tilde{x}$) is observed to exhibit stable behavior in particle-in-cell simulation codes even at very large Δt. To understand this, one begins by considering a possible remedy to the instability in a fixed well. For stable behavior one would want to use:

$$a_n = -\omega_0^2 x_n = -\omega_0^2(\tilde{x} + \Delta t^2 a_n/2), \tag{26}$$

so that $a_n(1+\omega_0^2 \Delta t^2/2) = -\omega_0^2 \tilde{x}$. Thus to regain stability one should replace, for any x, $a(x)$ by $a(x)/(1+\omega_0^2 \Delta t^2/2)$ in the particle advance. Then, folding in the effect of interpolating at $\tilde{x}$, one obtains

$$W_0^2 \Delta t^2 = \frac{\omega_0^2 \Delta t^2/(1+\omega_0^2 \Delta t^2/2)}{1-(\omega_0^2 \Delta t^2/2)/(1+\omega_0^2 \Delta t^2/2)} = \omega_0^2 \Delta t^2; \tag{27}$$

that is, the "approximate" dispersion relation with its global stability is recovered by an appropriate rescaling of the depth of the potential well.

Now consider what happens in a self-consistent direct-implicit plasma simulation code. If one assumes that $\omega_0 \approx \omega_p$ is the highest frequency, most "numerically dangerous" mode in the system, then the implicit field solver does indeed replace the electric field $E(x)$ by $E(x)/(1+\chi)$, where the "effective susceptibility" $\chi \equiv \frac{1}{2}\omega_p^2 \Delta t^2$ [6, 8]. The code computes the free-streaming charge density $\tilde{\rho}$ by accumulating contributions from particles at $\tilde{x}$; however, the restoring acceleration is not $-\omega_p^2 \tilde{x}$, but rather is $-\omega_p^2 \tilde{x}/(1+\chi)$; it has been reduced by just the factor needed to keep $W_0^2 \Delta t^2$ positive and equal to $\omega_p^2 \Delta t^2$. The field solver serves to prevent the mover's instability by adjusting the depth of the potential well. A similar effect should reduce the field magnitude in implicit codes based upon the "moment method" [1, 10]. As result, the net dispersion relation in an electrostatic, unmagnetized plasma simulation code (with interpolation at $\tilde{x}$) should most closely resemble the "approximate" relation first presented.

Thus interpolation at $\tilde{x}$ appears to be quite reasonable for a code in which the highest frequency oscillations are plasma waves. However, in codes which incorporate a fixed, non-self consistent field or a different set of equations, the system may admit modes with frequencies higher than ω_p, and reduction of E by the factor $(1+\chi)^{-1}$ may not suffice to keep $W_0^2 \Delta t^2$ positive. In such a case numerical instability may arise.

VII. Particle Mover Tests

The adjustable mover has been tested in a small code which advances test particles in a (frozen) power-law potential. In Fig. 3 the results of such a calculation

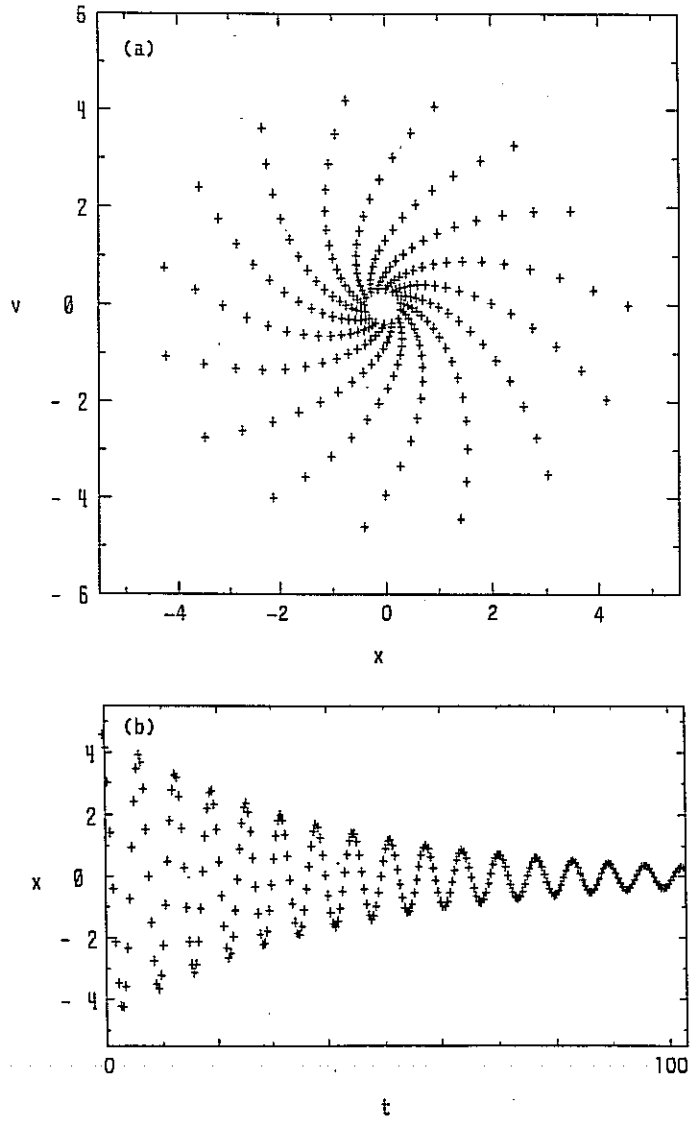

FIG. 3. Test of the $d1$ limit of the adjustable mover ($\theta = 1$) in a harmonic potential. (a) $x-v$ phase plane; (b) position vs time.

in a harmonic well are displayed. In these runs $x_0 = 4.6$, $\omega_0 = 1.0$, and $\Delta t = 0.4217$. Figure 3a is a sequence of points in the $x-v$ phase plane; the orbit begins at the point on the far right and spirals clockwise inward. Figure 3b is a time-history of the position. From figures such as these one can obtain the real frequency and the damping rate; these agree indentically (within very small measurement errors) with

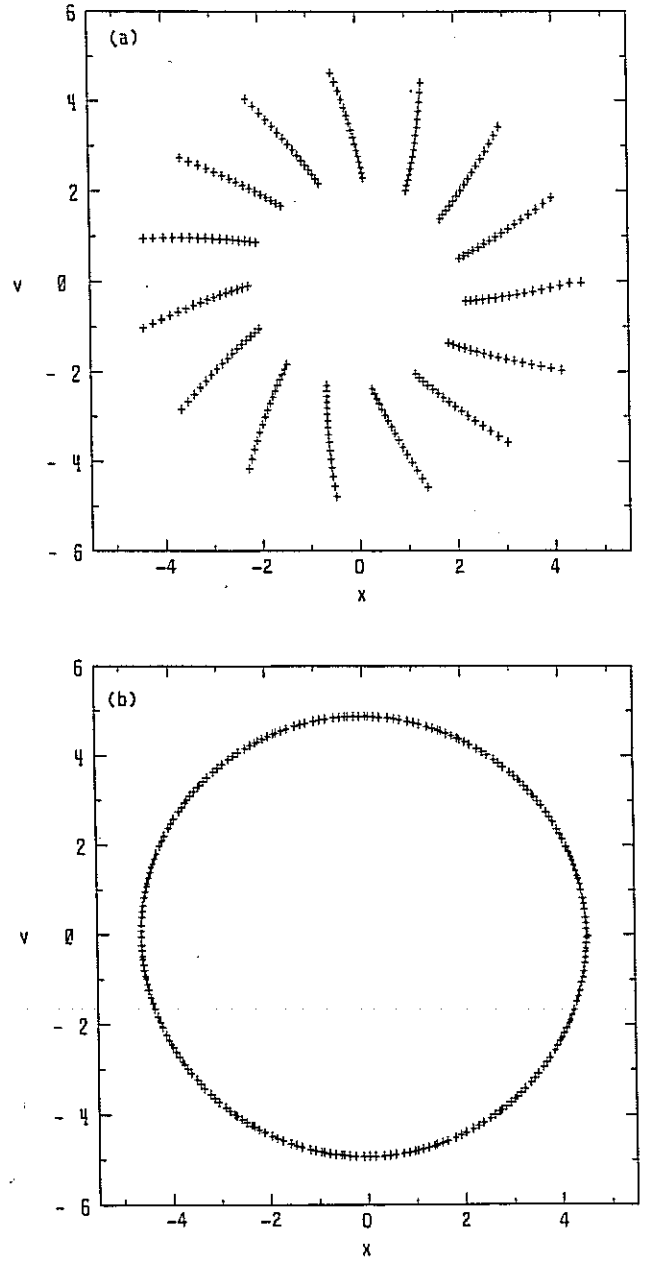

FIG. 4. Tests of the adjustable mover; $x-v$ phase plane for (a) $\theta=\frac{1}{2}$; (b) $\theta=0$.

the exact dispersion relation presented above. In Fig. 4a the phase-plane trajectory for $\theta = \frac{1}{2}$ is shown; the damping is clearly less rapid and the real frequency greater than for the $d1$ scheme. Figure 4b shows the undamped trajectory obtained when $\theta = 0$. Similar tests have confirmed the behavior for $\theta = 0$ and $\omega_0^2 \Delta t^2 = 4$, and the numerical instability of all schemes at larger timestep sizes.

VIII. Full Direct-Implicit Simulation Tests

The adjustable damping scheme has been implemented in a fixed step-size version of the Multiscale Implicit Simulation Testbed code (MIST) [9]. The code is one-dimensional and unmagnetized and uses "integral" centering of x and v. First, the cold plasma dispersion function (Eq. (17)) was verified with several tests. One such run employed 32 cells, 32 particles, $\Delta t = 10$, $\theta = \frac{1}{2}$, $\omega_{pe} = 1$, system length $= 1$, frozen ions, and the fundamental mode excited. For these parameters theory predicts $|z| = 0.7206$, $\omega_r \Delta t = 1.5467$. In Fig. 5 the field energy is plotted as a function of time. It falls 6.8 decades in 24 steps, thus the displacement δx falls by 3.4 decades, and $|z| = 0.7217$, well within the measurement error. The observed real frequency $\omega_r \Delta t$ is clearly near $\pi/2$; the "beat" (varying amplitude) in the figure indicates that the oscillation is not purely odd–even, in agreement with an analytical value $\lesssim \pi/2$.

Other tests examine free expansion of a plasma slab into vacuum [11]. Three

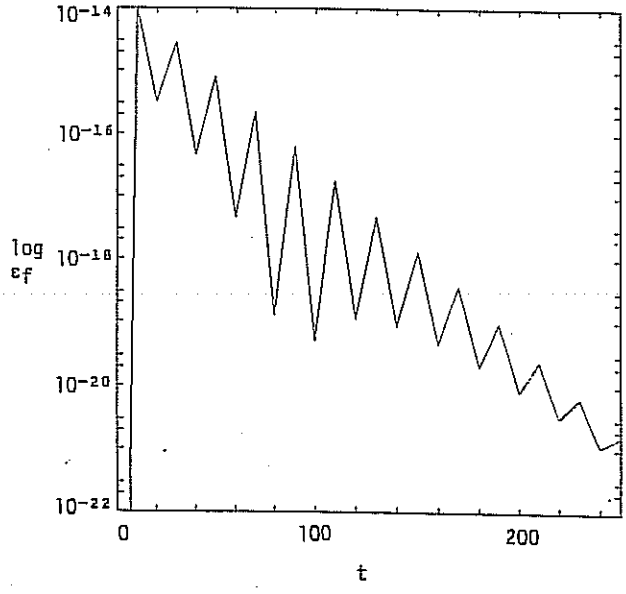

Fig. 5. Cold plasma dispersion test, field energy vs time for run PLAD.

runs, with differing values of the damping parameter θ, were carried out. The parameters were: system length $L=1$, $\Delta t=8$, $\omega_{pe}=1$, $m_i/m_e=900$; the run was carried out until $t_{max}=20{,}000$. There were 8192 particles of each species, 512 grid points, and the initial slab length was 0.25. The electron thermal velocity was $v_{th,e}=9.8\times10^{-5}$, that for the ions $v_{th,i}=1.03\times10^{-6}$.

In Fig. 6 are displayed the results of run 1, which used the $d1$ scheme ($\theta=1$). Figure 6a is a snapshot of the ion density as a function of position at $t=15{,}000$; it has been smoothed somewhat by averaging over 100 timesteps. The slab has expanded so that the right-hand leading edge of the expansion has moved from 0.625 to almost 0.8. Figure 6b is the time-history of the total energy; a rapid falloff from the initial value of 131 (normalized to 10^{-11}) to a minimum of 98 (at $t=8000$), followed by a rise to 110, can be seen. The electron kinetic energy (not shown) falls smoothly from 120 to 48 over the course of the run, while that of the ions rises from 12 to 61. It should be noted that the parameters for this run were deliberately chosen to give a major failure of energy conservation; the $d1$ scheme does not always do this poorly.

In Fig. 7 the results of run 2, with $\theta=0.6$, are displayed; in this run the leading edge is at or slightly beyond 0.8, and energy conservation is much improved: The minimum of 118 is reached at about $t=5000$, and the subsequent rise brings the total to 142. The electron kinetic energy falls to 65, the ion kinetic energy rising to 77.

For run 3 shown in Fig. 8, the mover was adjusted to give very little damping, $\theta=0.2$. The leading edge of the slab is at about 0.83, though the higher density regions of the expansion and the rarefaction wave still agree well with those of the

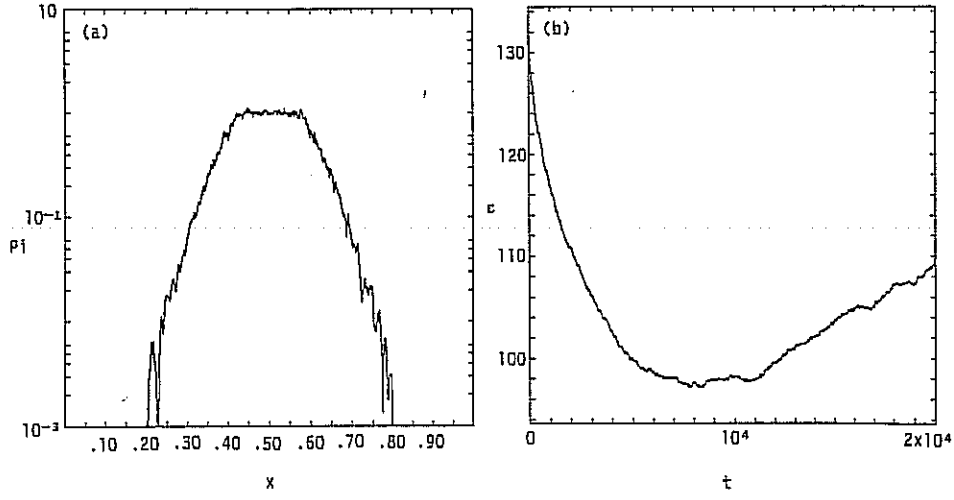

FIG. 6. Direct-implicit simulation test run 1 of plasma expansion into vacuum using adjustable mover with $\theta=1$ ($d1$ limit): (a) snapshot of ion density at $t=15{,}000$; (b) time history of total energy.

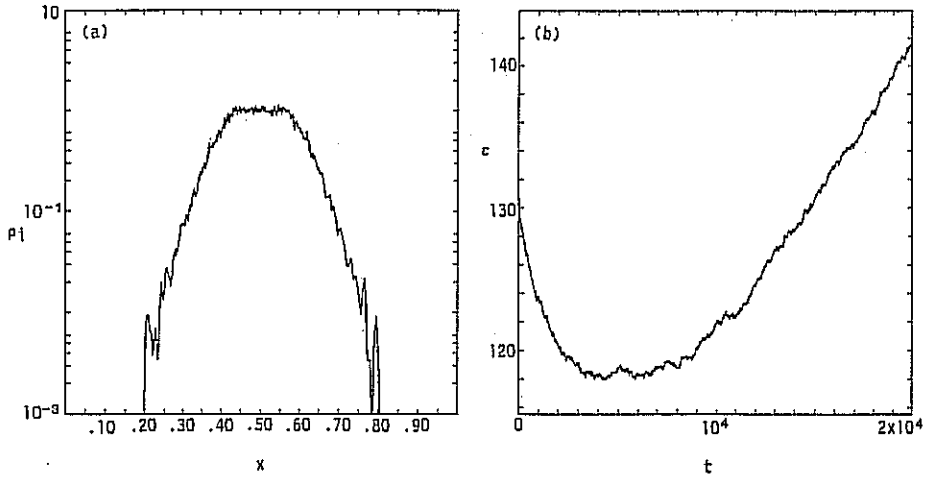

FIG. 7. As in Fig. 6, but run 2 using adjustable mover with $\theta = 0.6$.

above runs. Energy conservation is quite poor; the total grows from 131 to 212 over the course of the run. The electron kinetic energy falls only to 98, the ion kinetic energy rising to 113. It is expected that the centered scheme ($\theta = 0$) would heat even more rapidly. A larger timestep or thermal velocity would be expected to reduce the heating and make small values of θ the appropriate choice.

Of course, these diagnostics are quite crude, and energy conservation is only one very gross measure of overall accuracy. Nonetheless, it seems quite clear that the numerical damping imposed by the $d1$ scheme can be controlled by careful use of the adjustable mover.

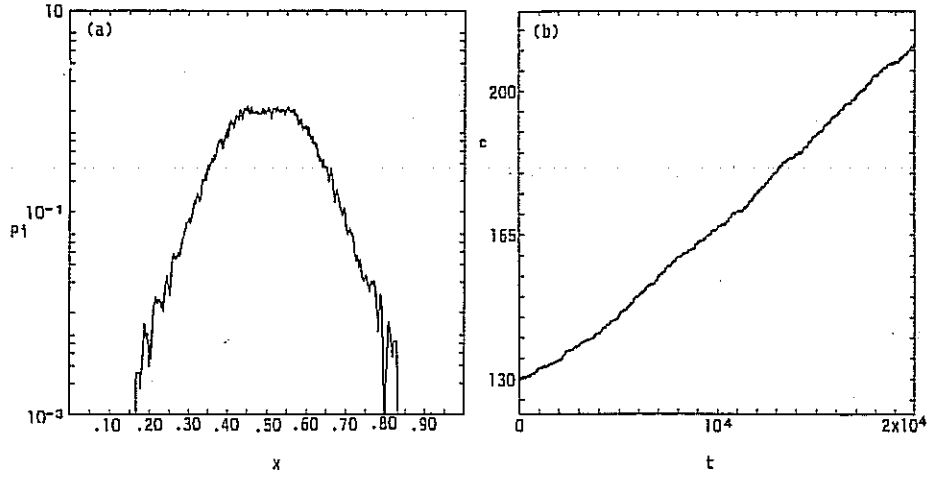

FIG. 8. As in Fig. 6 and 7, but run 3 with $\theta = 0.2$.

IX. Non-Constant Timestep and Damping Parameter

For flexibility it is desirable to be able to vary the timestep size over the course of a simulation, either globally or on a particle-by-particle basis in a full multiscale implementation [9]. It may also be desirable to alter the damping parameter θ used for each particle's trajectory calculation as the particle moves about in phase space. One might even want to use different values of θ for the different vectorial components of the acceleration, for some anisotropic problems. Here it is shown how the first two of these three objectives can be achieved while preserving time-centering so that the overall accuracy of the computation is not compromised. Further motivation for this topic is presented in the Discussion.

Our goal is to be able to double or halve the timestep size "between steps" in a scheme which advances x and v from $\mathrm{tl}\, n-1$ to $\mathrm{tl}\, n$. To do this it is necessary to modify the lag-averaged acceleration $\bar{a}$. A restriction to factor-of-two changes in Δt is imposed because (1) it is algorithmically convenient, and (2) more extreme changes would generally result in the greater accuracy of the smaller step being wasted because of large errors in the adjacent larger step. It is still possible to find a step size within a factor of $\sqrt{2}$ of any desired target size.

First consider halving the step size, from Δt to $\delta t = \Delta t/2$. Before taking a step at the new step size, compute

$$\bar{a}'_{n-1/2} = a_n/2 + \bar{a}_{n-1}/2 \tag{28}$$

and replace $\bar{a}$ in the particle array with this quantity. This "more recent" lagged average is defined at the appropriate time level for the new, smaller step to come. It is straightforward to verify that for $\theta = 0$, $\bar{a}'_{n-1/2} = \bar{a}_{n-1} = a_n$, while for $\theta = 1$, $\bar{a}'_{n-1/2}$ is a re-lagged average as expected.

Similarly, one can double the step size to $Dt = 2\,\Delta t$, using

$$\bar{a}'_{n-2} = 2\bar{a}_{n-1} - a_n, \tag{29}$$

and replace $\bar{a}$ with this quantity. For $\theta = 0$, $\bar{a}'_{n-2} = \bar{a}_{n-1} = a_n$, while for $\theta = 1$, $\bar{a}'_{n-2} = \bar{a}_{n-2}$.

If one wishes to change the damping parameter θ to a new value (call it ϕ), this can be accomplished while preserving the time-centering. One must replace the lag-averaged electric acceleration $\bar{a}_{n-1}$ with a new value, denoted by $\bar{b}_{n-1}$. As an ansatz, let

$$\bar{b}_{n-1} = \gamma \bar{a}_{n-2} + (1-\gamma)\,\bar{a}_{n-1}. \tag{30}$$

Equating time levels (see Eq. (6)), one has

$$n - (\phi/2)/(1-\phi/2) = \gamma[n-1-(\theta/2)/(1-\theta/2)] + (1-\gamma)[n-(\theta/2)/(1-\theta/2)] \tag{31}$$

or

$$\gamma = (\phi/2)/(1-\phi/2) - (\theta/2)/(1-\theta/2). \tag{32}$$

A desirable property of these operations is that they commute. That is, halving the step size and then doubling θ is equivalent to performing the same operations in the reverse order.

X. Discussion and Directions for Future Research

Implicit PIC codes which employ damped equations of motion should not be expected to conserve energy. Nonetheless, gross non-conservation of energy is likely at the very least to make results difficult to interpret. In practice, one observes reasonable energy conservation in a narrow valley of parameter space. That is, it is desirable to retain, in the absence of spatial filtering, the relation $v_{\text{thermal}}\,\Delta t/\Delta x \approx 0.3 \pm 0.1$ [3]. Larger timesteps lead to excessive damping, smaller ones to heating due to finite-grid instability (on a mesh with $\Delta x \gg \lambda_d$) [13]. The timestep size, thermal velocity, mesh size, amount of spatial filtering, initial loading scheme (e.g., a quiet start), and number of simulation particles all affect the heating rate.

An important consideration in practical particle-in-cell plasma simulation is the prescription by which grid data are interpolated to the particle positions and by which particle charge and current are deposited on the mesh. The dispersion relations obtained in Sections V and VI, with interpolation of the field at locations x_n and $\tilde{x}_n$ respectively, differ qualitatively. In this paper, grid effects have been ignored in the dispersion analysis, and the spatial interpolant has been assumed "exact." Not explicitly discussed herein is the utility of high-order splines and/or strong spatial smoothing. Such techniques may ameliorate grid aliasing and attendant errors in (e.g.) energy conservation, which are otherwise present when the mesh spacing is much larger than a Debye length (as is often the case in implicit simulations). A hazard that arises when one employs a damped algorithm is that one may operate in a regime where the finite-grid instability is strong and yet may obtain reasonable energy conservation by use of much damping. Under such circumstances even large-scale physics might be poorly represented. A topic for future research is the interplay of the spatial interpolant and the particle and field advances described herein.

The simple tests described herein show that the adjustable mover does in fact work as expected. Its simplicity renders it easily retrofitted into any simulation code using the $d1$ mover with either "integral" or "staggered" centering. Indeed, implementation in MIST (integral centering) was completed in less than a half hour and only a few lines of code were changed. Having an extra "knob to tune" is a double-edged sword, as this increases the dimension of the parameter space. Nonetheless, the new mover should provide access to a wider range of problems than was previously possible.

It is somewhat surprising that the $d1$ mover, as it is commonly implemented, admits instability in a fixed well at values of the timestep which are quite moderate. The restriction that trapping oscillations should be resolved [12] is related to this

stability constraint. The reasons for the mover's apparent stability in full simulation codes are thus somewhat subtle. One should beware any fixed (externally specified) fields whose magnitude is not subject to control by the self-consistent fieldsolver and be careful to avoid violating the trapping frequency constraint.

The new adjustable mover does not significantly extend the regime of numerical stability, but the analysis presented here suggests several possibilities. To suppress the instability, were it to be observed in practice, one might enlarge χ beyond $\frac{1}{2}\omega_p^2 \Delta t^2$ (taking into account externally imposed oscillations, not just plasma oscillations); this is a variation of the "remedy" described above. Alternatively, the force interpolation might be altered so that it used the field at the future position; this might require iteration. Finally, a timestep control might be employed, either globally (where practical) or in a multiscale algorithm.

Explicit schemes with tunable damping properties similar to those of the implicit algorithms presented here are readily obtained by extrapolating the electric component of the acceleration forward (along the trajectory) by one timestep. Thus one replaces a_{n+1} by $2a_n - a_{n-1}$ and the particle advance no longer involves the field at the future time level. Of course, the dispersion relation differs from that of the implicit advance, and large-timestep stability cannot be obtained in this manner; in fact, the stability constraint is slightly more severe than it is for the leapfrog advance. Nonetheless, the noise-reduction aspects of the implicit advance are preserved at small timestep. Indeed, the asymptotic damping behavior (but not the real frequency shift) is identical. The real frequency approaches, for small timestep:

$$\omega_r/\omega_0 = 1 - \omega_0^2 \Delta t^2 [\theta/4(2-\theta) - 1/24] + \cdots. \tag{33}$$

The scheme reverts to the leapfrog advance when the damping parameter θ is set to zero; this is reflected in the above relation. Such schemes may have real utility because they are simpler than implicit methods, yet should yield quieter simulations than undamped methods.

An implicit scheme similar to the $d1$ algorithm is used to advance the electromagnetic field in Ref. [5]. It is straightforward to apply the adjustable implicit algorithm to that problem as well. Langdon and Barnes have previously suggested a different variably-damped implicit algorithm (a blend of the $d1$ and leapfrog schemes) for the EM advance [8]; they note that the $d1$ fraction could be set nonzero only where cells were small (to afford stability when the leapfrog Courant condition on light wave propagation is violated locally). The undamped leapfrog advance could be used over the remainder of the mesh; by operating near the stability boundary, improved dispersion might be obtained. Tests of their scheme; and of the one derived from the particle mover of this paper, are topics for future study.

An adjustably-damped *explicit* EM advance is possible, and may be useful in removing extraneous field noise in electromagnetic codes. An easily implemented explicit scheme is

$$\mathbf{E}_{n+1}=\mathbf{E}_n+c\,\Delta t\,\nabla\times\mathbf{B}_{n+1/2}-4\pi\,\Delta t\,\mathbf{J}_{n+1/2}; \tag{34a}$$

$$\mathbf{B}_{n+3/2}=\mathbf{B}_{n+1/2}-c\,\Delta t\,\nabla\times\left[\left(1+\frac{\theta}{4}\right)\mathbf{E}_{n+1}-\frac{1}{2}\mathbf{E}_n+\left(\frac{1}{2}-\frac{\theta}{4}\right)\bar{\mathbf{E}}_{n-1}\right], \tag{34b}$$

where

$$\bar{\mathbf{E}}_{n-1}=\left(1-\frac{\theta}{2}\right)\mathbf{E}_n+\frac{\theta}{2}\bar{\mathbf{E}}_{n-2}. \tag{35}$$

On a one-dimensional spatial lattice with central differencing, a dispersion relation for a mode with spatial wavenumber k can be obtained. Defining $\Omega\equiv(c\,\Delta t/\Delta x)\sin(k\,\Delta x/2)$, one finds

$$\sin^2\left(\frac{\omega\,\Delta t}{2}\right)=\Omega^2\left[1-\frac{2\theta\sin^2(\omega\,\Delta t/2)}{2e^{-i\omega\,\Delta t}-\theta}\right]. \tag{36}$$

For small $\omega\,\Delta t$, the damping is $\gamma\,\Delta t\approx -8\theta/(2-\theta)^2\,\Omega^4$, and so $\gamma\propto(k\,\Delta x)^4$. This class of methods is under active investigation, by itself and in combination with the explicit damped particle advance mentioned above. The results of further analysis and tests will be described in a future paper [14].

Appendix

Here the integral-level $d1$ scheme is derived from the staggered one. For the staggered scheme, the final-push is

$$\bar{a}_{n-1}=\tfrac{1}{2}[\bar{a}_{n-2}+(q/m)\,E_n(\tilde{x}_n)]$$
$$v_{n-1/2}=\tilde{v}_{n-1/2}+(\Delta t/2)(q/m)\,E_n(\tilde{x}_n)$$
$$x_n=\tilde{x}_n+(\Delta t^2/2)(q/m)\,E_n(\tilde{x}_n).$$

Then, the pre-push is

$$\tilde{v}_{n+1/2}=v_{n-1/2}+(\Delta t/2)\,\bar{a}_{n-1}$$
$$\tilde{x}_{n+1}=x_n+\Delta t\,\tilde{v}_{n+1/2}.$$

One moves the computation of $\tilde{x}_{n+1}$ to the beginning of the final push, where it becomes $\tilde{x}_n=x_{n-1}+\Delta t\,\tilde{v}_{n-1/2}$. One then *relabels* $\tilde{v}_{n+1/2}$, calling it v_n (it is formally centered at tl n, so this is perhaps a notational improvement). Thus:

$$v_n=v_{n-1/2}+(\Delta t/2)\,\bar{a}_{n-1}$$
$$=v_{n-1}+(\Delta t/2)\,\bar{a}_{n-1}+(\Delta t/2)(q/m)\,E_n(\tilde{x}_n).$$

Explicitly, the integral-level $d1$ algorithm final-push is:

(1) $\tilde{x} = x_{n-1} + \Delta t\, v_{n-1}$

(2) $a = (q/m)\, E_n(\tilde{x})$ (interpolation of field from mesh)

(3) $\bar{a}_{n-1} = \frac{1}{2}[a + \bar{a}_{n-2}]$

(4) $v_n = v_{n-1} + (\Delta t/2)\, \bar{a}_{n-1} + (\Delta t/2) a$

(5) $x_n = \tilde{x} + (\Delta t^2/2) a.$

The pre-push to the next time level is

(6) $\tilde{x} = x_n + \Delta t\, v_n.$

(7) Using this new $\tilde{x}$, compute ρ and χ for the field solver.

As a further step, one can interchange steps (3) and (4), replacing $\bar{a}_{n-1}$ by its definition in terms of a and $\bar{a}_{n-2}$:

(3′) $v_n = v_{n-1} + (3\,\Delta t/4) a + (\Delta t/4)\, \bar{a}_{n-2}$

(4′) $\bar{a}_{n-1} = \frac{1}{2}[a + \bar{a}_{n-2}].$

This form of the $d1$ scheme is clearly the $\theta = 1$ limit of the adjustable mover.

Acknowledgments

The author has benefitted from useful discussions with J. Ambrosiano, S. T. Brandon, B. I. Cohen, D. W. Hewett, A. B. Langdon, D. E. Nielsen, Jr., S. E. Parker, P. Rambo, and S. L. Ray. This work was performed under the auspices of the U.S. Dept. of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-ENG-48.

References

1. R. J. Mason, *J. Comput. Phys.* **41**, 233 (1981).
2. B. I. Cohen, A. B. Langdon, and A. Friedman, *J. Comput. Phys.* **46**, 15 (1982).
3. B. I. Cohen, A. B. Langdon, D. W. Hewett, and R. J. Procassini, *J. Comput. Phys.* **81**, 151 (1989).
4. B. I. Cohen, M. E. Stewart, and C. K. Birdsall, "Direct Implicit Particle Simulation of Tandem Mirrors," 11th International Conference on Numerical Simulation of Plasmas, Montreal, Canada, June 25–27, 1985.
5. D. W. Hewett and A. B. Langdon, *J. Comput. Phys.* **72**, 121 (1987).
6. A. Friedman, A. B. Langdon, and B. I. Cohen, *Comments Plasma Phys. Controlled Fusion* **6**, 225 (1981).
7. D. C. Barnes, T. Kamimura, J.-N. LeBoeuf, and T. Tajima, *J. Comput. Phys.* **52**, 480 (1983).
8. A. B. Langdon and D. C. Barnes, "Direct Implicit Plasma Simulation," in *Multiple Time Scales*, edited by J. U. Brackbill and B. I. Cohen (Academic Press, Orlando, FL, 1985), p. 335.
9. A. Friedman, S. E. Parker, S. L. Ray, and C. K. Birdsall, "Multi-Scale Particle-in-Cell Plasma Simulation," to appear in *J. Comput. Phys.*

10. J. DENAVIT, *J. Comput. Phys.* **42**, 337 (1981).
11. J. DENAVIT, *Phys. Fluids* **22**, 1384 (1979).
12. A. B. LANGDON, B. I. COHEN, AND A. FRIEDMAN, *J. Comput. Phys.* **51**, 107 (1983).
13. C. K. BIRDSALL AND A. B. LANGDON, *Plasma Physics Via Computer Simulation* (McGraw–Hill, New York, 1985), p. 179.
14. Further development of the explicit methods is being carried out in collaboration with J. Ambrosiano, S. T. Brandon, D. E. Nielsen, Jr., and P. Rambo.

Printed by Catherine Press, Ltd., Tempelhof 41, B-8000 Brugge, Belgium